VFAST Transactions on Software Engineering
http://vfast.org/journals/index.php/VTSE@ 2020 ISSN(e): 2309-3978;ISSN(p): 2411-6246
Volume 8, Number 1, January-December, 2020

pp. 01-07

ATTACK DETECTION FROM NETWORK TRAFFIC USING MACHINE LEARNING

Maryum Nawaz, Muhammad Arsalan Paracha, Dr. Abdul Majid, Dr. Hanif Durad
Department of Computer and Information Sciences
Pakistan Institute of Engineering and Applied Sciences
Islamabad, Pakistan
maryum.nawaz794@gmail.com, arsalanparacha@gmail.com, [abdulmajiid; hanif] @pieas.edu.pk

Abstract— Network Security Management is not only becoming difficult but also becoming impossible as size of networks grow. Attacks grow beyond the current ability of security related management tools to identify the attacks and respond quickly. So a machine learning based model is designed to detect most recent and up to date attacks from network flow data of network devices, i.e. network switches, routers, wireless routers, firewalls, etc. Extreme Gradient Boosting based model is designed for attack detection that provides 91.61% detection rate, generate very few false alarms at rate of 0.005% and misses attacks at 8.38% rate over CICIDS dataset which recent open source dataset containing network flow information of network devices.
Keywords— Network flow, XGBoost, CICIDS2017, Denial of Service

1. Introduction. Network Security Management is not only becoming difficult but also becoming impossible as size of networks grow. Also it is becoming an essential element of organizational processes. Increase in attacks over network is because of the intellectual challenge of attackers or hackers and also because of the increasing amount of ransoms. Also, the attacks grow beyond the current ability of security related management tools to identify the attacks and respond quickly. If different methods of attack applied and eventually prevented, the attackers try new approaches with more intelligent features of attack. So, maintaining and management of network Security is ever changing and on-going process.

Attacks on computer networks can be of different forms, and each attack can involve many types of security events. Security incidents include stealing someone's private information like passwords, credit card information, or denial of service by using different mechanisms like Rootkits, Trojans, viruses, worms, traffic bombardment and redirecting communication etc.

Each device maintains its flow of data which is known as network flow that contains all information regarding its traffic. Flow data represents a single packet flow in the network with the identification of 5-tuples i.e. IP address of destination host, IP address of source, destination port, source port and protocol. Based on this, packets are aggregated into flow records that accumulate the amount of transferred data, the number of packets and other information from the network and transport layer. Basically a single flow record contains multiple number of packets for connection with some node for certain amount of time. This data can be passed to some collector in form of flow records which perform analysis over that data. Network Flows of different devices can also be helpful in monitoring purpose in case of network devices as it contains all necessary information regarding network passes through these devices and maintain all flow records of data and packets transferred between two nodes.

To apply machine learning (ML) techniques a standard dataset is needed. CICIDS2017 [5] is used for the purpose of designing the model to monitor network devices and generate alerts for different incidents or attacks. This is most recent openly available labelled dataset which helped in training ML model in better way. CICIDS dataset has benign and most recent 14 kinds of attacks that depicts real-world data. The dataset is of huge volume about 11.5 GB of data containing more than 2.2 Million of instances of 83 different network flow features.

Khairaisat et al. surveyed different intrusion detection datasets and discuss different ML techniques used for these datasets in their research paper [1]. They classify these datasets on the basis of attacks. Ranjit et al. [2] evaluated CICIDS2017 dataset. CICIDS2017 cater all recent attacks and features which has not been addressed by other older datasets which is plus point of this dataset and attract researchers towards themselves. But still there are some shortcomings like dataset is highly imbalanced, there are missing values in the dataset, its volume is very huge and data is not present in single file. After describing shortcomings Ranjit et al. present possible solutions for handling class imbalance problem which is merging classes of same kind into one to avoid imbalance to some extent.

Boukhamla et al. [3] improve the performance of CICIDS dataset by applying different preprocessing techniques over dataset. After preprocessing over dataset they applied KNN, decision tree and Naïve Bayes and evaluate the performance of these algorithms in terms of detection rate and false alarm rates. They concluded that KNN perform better than other two algorithms in terms of considered performance measures. Usteby et al. in their research paper [4] proposed deep learning and random forest based model to recursively eliminate features from CICIDS2017 dataset and proved that there model improved the performance of different algorithms. Ahmim et al. [6] in their paper proposed a decision tree and rule based ML model to detect attacks from CICIDS2017 dataset and made comparison of results with other ML models and proved that their designed model performed better than other models.

In this paper Extreme Gradient Boosting (XGBoost) based classifier is proposed to detect most recent attacks using standard dataset named CICIDS2017. XGBoost is ensemble ML algorithm that improves and boost the weak algorithms to perform better. Model is first trained using 5-fold cross validation and parameters are optimized for better results. Classification results of different performance metrics which are specific to attack detection and attack misses are calculated. These performance measures are then evaluated and compared with other technique proposed in different research papers. An improved performance has been observed on XGBoost based classification than other techniques proposed by researchers.

Further sections of paper are organized as: section II provide dataset brief information and proposed methodology followed by Section III that will discuss the performance measures and give comparison of results. Finally section IV conclude the research paper.

2. Dataset And Methedology. Most recent and available open source dataset named CICIDS2017 is selected for detection of attacks from network devices information. This dataset is based on network flow information that contains almost each and every information about network devices traffic. Then XGBoost results are optimized over this dataset using different data preprocessing and data training techniques for better performance while testing.

- **. Dataset Description**. CICIDS2017 dataset is generated from pcap files generated for traffic of organization using CICFlowmeter. There are 83 different features, some are statistical features (mean, median, standard deviation, min, and max) such as Number of packets, Duration, Length of packets, number of bytes, bulk rate and Number of segments in both directions direction i.e. forward and backward. Then features related to different flags acknowledgement, push, reset, sync are calculated separately. The output generated in CSV format. Data has six columns which identified each flow, namely FlowID, Destination IP, Source IP, SourcePort, DestinationPort.

  Data generated is collected from 25 multiple hosts including victim computers, attack sources, servers, routers, switches and firewall. The main properties of dataset is described by Sharafaldin et al. [5] which make this data distinguishable. These properties contains complete configuration of network, dataset is labelled and contains 15 different classes including Benign, FTP-Pataor, SSH-Patator, Bot, XSS, SqlInjection, Portscan, DDoS, Dos-Hulk, Dos-Slowloris, Dos-SlowHttp, Dos-Goldeneye, Heartbleed, Infiltration and Brute-Force [5]. Fig. 1 shows the whole distribution of data with reference of assigned classes.

- ***Methedology.*** As CICIDS2017 dataset is of huge volume and contains millions of instances so applying machine learning algorithms without preprocessing may lead to inefficiency. To overcome this problem in followed methodology different filtering and preprocessing techniques are applied over CICIDS dataset as data optimization techniques.

- **Data Filtering/Cleaning.** *Given dataset has some cells contains NaN or infinity values. So the rows containing these values are deleted, as a result 2600 rows were filtered out. Some columns named FlowID, Source IP, Destination IP and Timestamp contains string values which is not handled by ML algorithms, and also these columns are not related to assigned classes.*

3. Data Normalization. In preprocessing, as whole data is spread over wide range and there exist large variation of data values as shown in fig. 2 that plot values of two features of dataset. Due to this widespread various ML

algorithms could not perform accurately because of their range. So data is preprocessed in a way that normalization and scaling of data is applied.

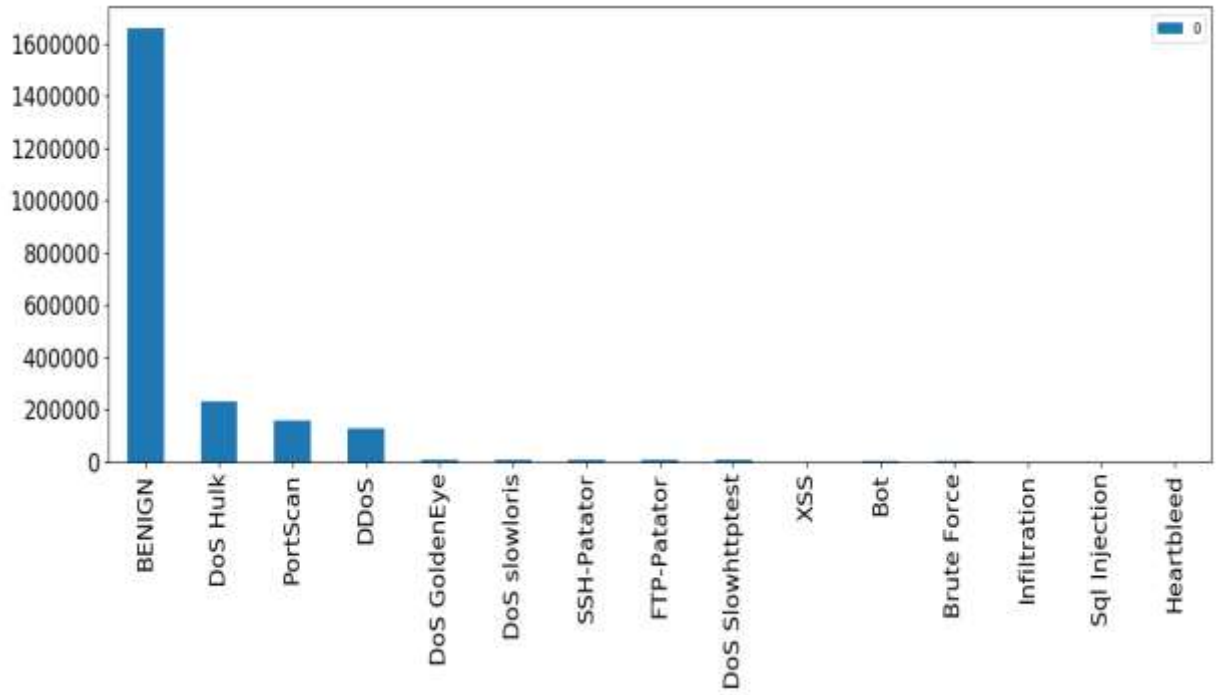

Fig. 1. Data Distribution of CICIDS dataset

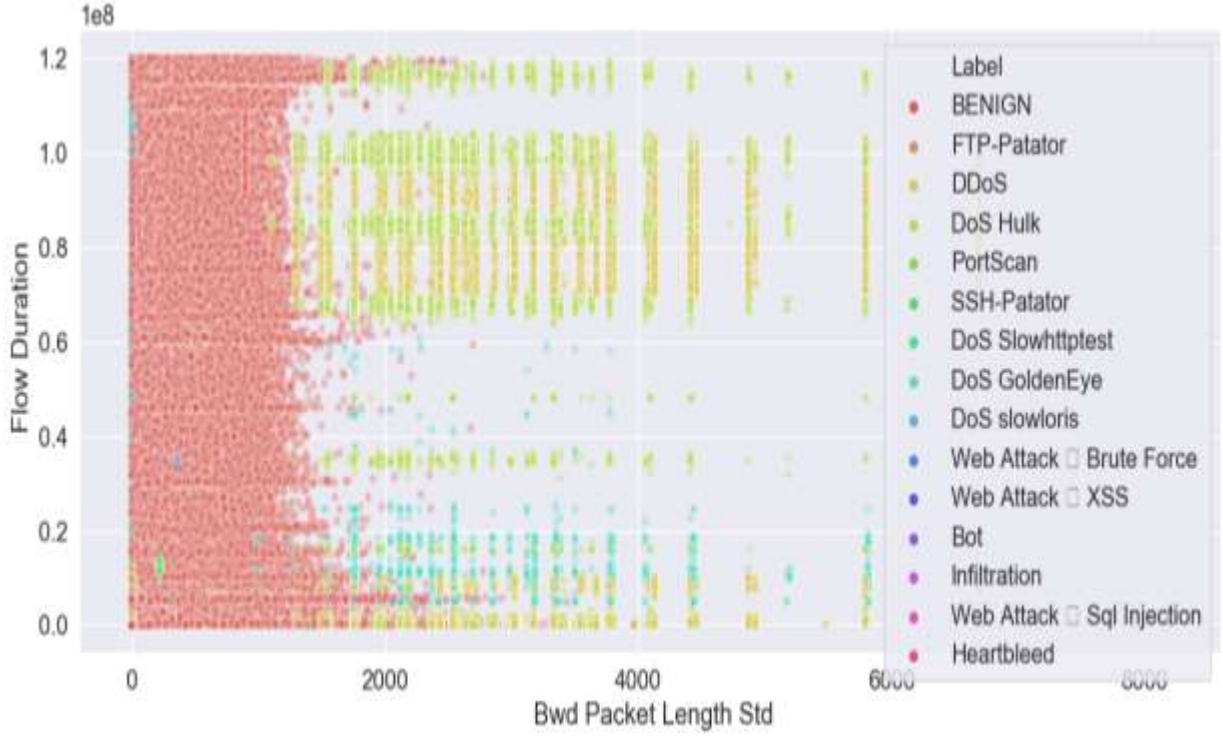

Fig. 2 Data plot for Flow Duration and Backward Packet Length

TABLE I. Merged Classes Based on Solution Presented by Ranjit Et Al. [2]

Sr. No	New Labels	Old Labels	Number of Instances
1	Benign	Benign	2210787
2	Bot	Bot	1966
3	Brute Force	FTP-Patator, SSH-Patator	13835
4	Dos/DDos	DDoS, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed	294506
5	Infiltration	Infiltration	36
6	PortScan	PortScan	158930
7	Web Attack	Brute Force, Sql Injection, XSS	2180

- ***Data Preprocessing***. It can be seen from Fig. 1that there is highly imbalanced ratio of different classes. Benign class

cover about 85% of whole data and contain more than 1.6 million instances from total of 2.2 million+ instances. While some classes like infiltration, sql-injection and heartbleed has less than 100 instances in data of millions. To avoid this problem a solution is presented in paper by Ranjit et al. [2] is applied over data and new classes distribution in dataset are shown in Table 1.

Now the number of classes reduced to 7 from 15. All kinds of Denial of Service (DoS) attacks generated using different tools are labelled as DoS or DDoS attacks, brute force attacks targeting different platforms are labelled as brute force attack and different kinds of websites or web servers attacks are counted as web attacks. New distribution of data after merging attack is shown in Fig. 3. As we can see imbalanced ratio is reduced to some extent.

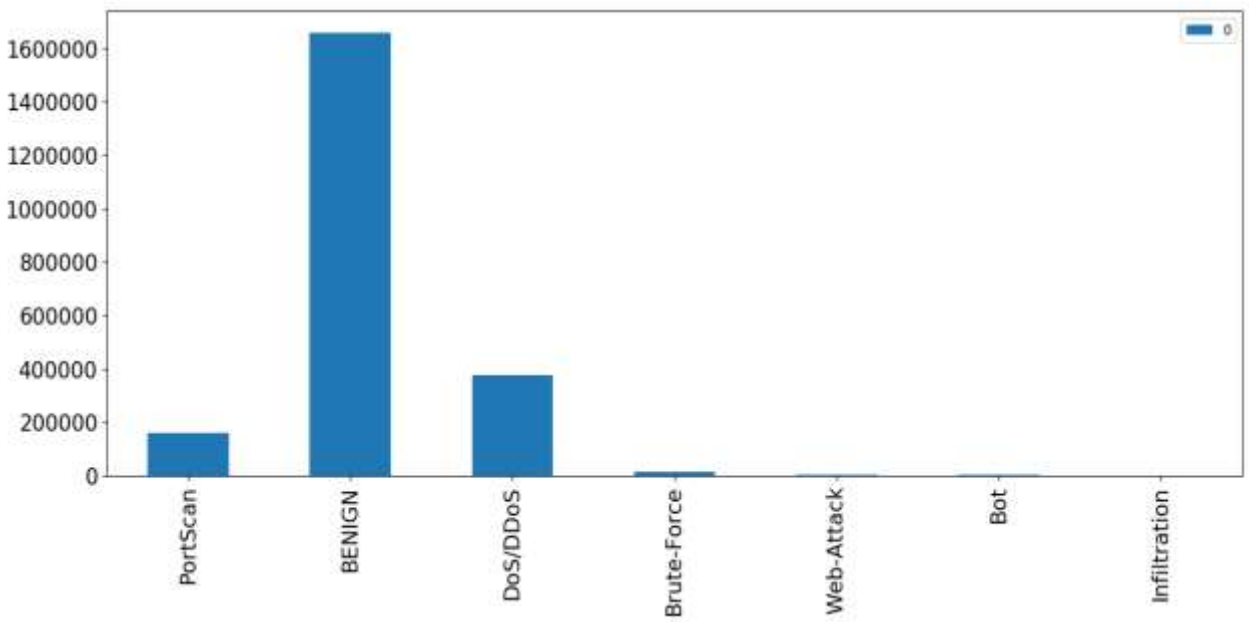

Fig. 3 New Data Distribution of CICIDS dataset

- **Training and cross validation** After initial data processing whole dataset is divided into 80-20 ratio for training and testing purpose. To avoid and cater biasing in results test data is kept totally hidden at time of training. To optimize and tune parameters of ML classifiers k-fold cross validation (CV) is applied over whole. Cross validation results are calculated over 3, 5, 7 and 10 folds and at the end optimized results obtained over 5-folds. In cross validation whole data is divided into k number of folds in which 1 fold is used as validation data while (k-1) folds are use as training data, and process repeated k number of times and in every iteration different folds are taken as validation set. So optimization is achieved using 5-fold CV parameters of ML classifier.

- **Testing over unseen data.** After getting optimized parameters from CV, test data is evaluated and performance measures are calculated over those set of parameters. For the purpose totally unseen data is tested to get unbiased results. At the time of testing performance of different measures reduced because of unseen data. Then test results are compared with ML algorithm results presented in other research paper for analysis purpose.

4. Results and Discussion. . An optimized and efficient ML algorithm is required for detection of attacks from network devices information. Training is performed using 5-fold cross validation. Multiple runs of different ML algorithms (KNN, RF, MLP, and XGBoost) over different number of folds with same set of parameters and in the end number of folds with best results among all was selected.

- Performance metrics. Performance metrics are calculated to evaluate the performance of any ML algorithm and to find out how effective is this algorithm. In attack detection main focus is to maximize the attack detection rate and to minimize the attacks miss rate. As positive samples are point of interest so performance measures taking positive samples as important factor are considered.

- ***Confusion Matrix:****Confusion matrix provide us detailed information about the data predications. It is not itself a performance measure but TP, FP, TN and FN from confusion matrix which can be calculated which are basis of all performance metrics. TP is true Positive which is true attacks detected, FP is False Positive which is benign or normal instance which is detected as attack that is false alarm. TN is true negative that is benign instance detected as benign one and FN is False Negative i.e. attack instance id detected as normal or benign one. Basic Structure of confusion matrix is shown in Table 2.*

TABLE II. **Confusion Matrix**

		Predicted	
		Positive (1)	***Negative (0)***
Actual	Positive (1)	True Positive	False Positive
	Negative (0)	False Negative	True Negative

a. Recall. *It basically defined the sensitivity or true positive rate (TPR) of algorithm i.e. at what ratio the algorithm find out true positive samples from whole all possible positive samples also called detection rate. The instance of interest are positive sample as in current scenario attacks are positive samples. Its value ranges from 0-1. But normally it is represented in percentage. For good algorithm this value should be larger. It is defined by formula:*

$$Recall = TPR = \frac{TP}{TP+FN} \quad (1)$$

b. Precision. *Precision defined as the positive predictive value. It is basically the number of positive sample detected from all data. Its value also ranges from 0-1 but represented in percentage. It is defined by formula:*

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

i. F1_score

It is harmonic mean of recall and precision. It is sensitive to extreme values. Its values vary from 0 to 1. 0 is worst value and 1 is the best score. Its formula is:

$$F1_{score} = \frac{2*TP}{2*TP+FP+FN} \quad (3)$$

ii. False Positive Rate

It is also defined as false alarm rate. It is basically the ratio of negative samples that has been misclassified as positive and generate false alarm of attack detected. This value should be smaller. Formula for FPR is:

$$FPR = \frac{FP}{FP+TN} \tag{4}$$

iii. False Negative Rate:

It is also defined as miss rate. It is basically the ratio of positive samples that has been misclassified as negative one. This value should be smaller. Formula for FNR is:

$$FNR = \frac{FN}{FN+TP} \tag{5}$$

Training Results. As described 5-folds cross validation is performed in order to tune parameters and train data. Different run of algorithms are applied over whole data with 5-folds each time with change in parameter values. Results are computed by taking average of results of each fold. Random search of different parameters is applied over validation for parameter tuning. As XGBoost is tree based ensemble algorithm so for XGBoost number of estimators, max depth of tree, number of columns in each tree and in each node are important parameters to tune. Most optimized results of XGBoost are obtained when number of estimators are 250, with maximum depth of tree is set to 7.

At optimized parameters set detection rate of attack was 96.16%, 99.48% of precision value, F1_score value was 0.9866, miss rate or false negative was at rate of 7.35% and false alarms were generated at the rate of 0.003%. These results obtained when algorithm is executed for 57.87 seconds.

Test Results. XGBoost based model was trained over optimized set of parameters and then 20% data, separated for testing purpose, was tested over that model. As unseen data is tested so performance of algorithm reduced to some extent. Now detection rate became 91.61%, precision value maintained to 99.71%, F1_score give score of 0.9403, false alarms generated at the rate of 0.005% and algorithm misses the attacks instances at the rate of 8.38%.

Fig.4 shows the graph that represents the values of precision against recall at different thresholds of each class of data for XGBoost algorithms. As seen from figure that for each class improved values of precision and recall are obtained.

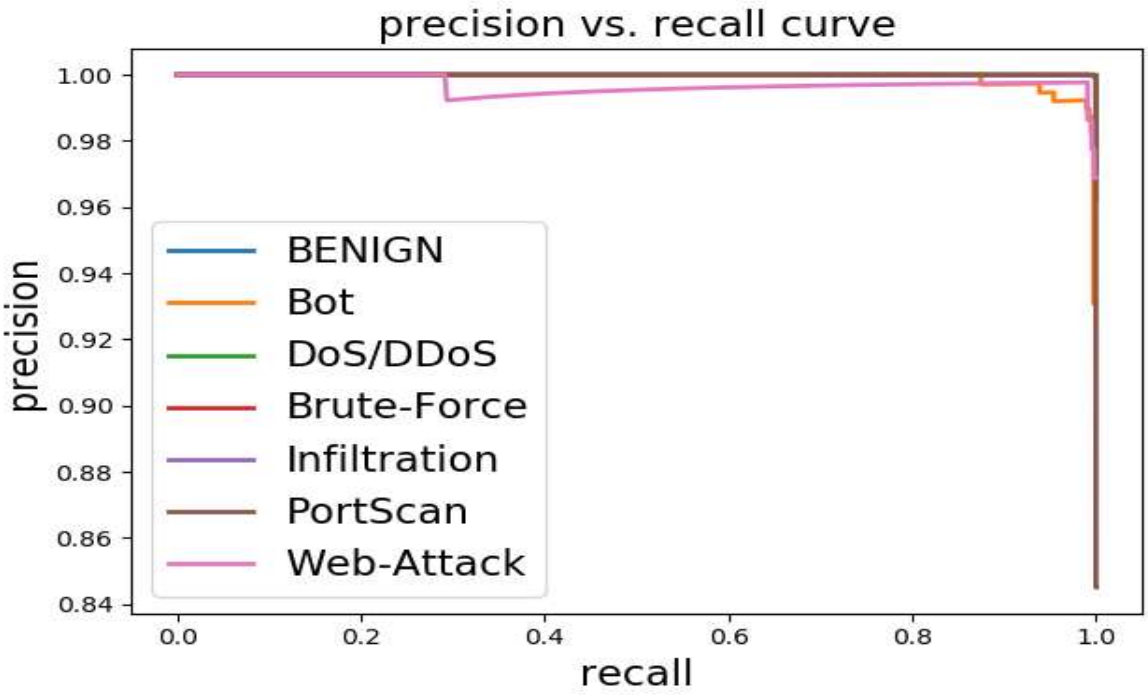

Fig. 4 Precision Recall Curve of XGBoost Results

Comapritve analysis. It is observed from literature that multiple studies and algorithms are proposed for CICIDS dataset. So test results of proposed model is compared with other algorithms presented in different research papers are given in Table III.

TABLE III. Performance Comparison of other Classifiers and Proposed Model

Models		**Performance Measures**					
	Ref	***Recall (%)***	***Preci-sion (%)***	***F1_ Score (%)***	***FPR (%)***	***Accu-racy (%)***	***Area Under ROC (%)***
DMLP	[4]	-*	-*	-*	-*	89	97.1
Ahmim et al.	[6]	94.47	-*	-*	1.15	96.66	-*
KNN	[3]	72.88	-*	-*	0.35	-*	-*
XG-Boost	**Prop-osed**	91.61	99.71	94.03	0.005	99.97	100

* Performance measures that has not been considered by researchers

All researchers consider different performance measures to calculate performance of their model. Results of all performance measures are calculated for proposed model in order to make comparison. It can be seen from Table III that proposed model performed better than all other model presented in different researches in terms of detection rate, precision, f1_score, false alarm rate, accuracy and are under ROC curve. Detection rate is slightly less than proposed model of Ahmim et al. [6] which is based on decision trees and rule based model but still false alarm are generated much less than the Ahmim model and also accuracy measure performed better than that model.

5. Conclusion. This paper presented the XGBoost based model for attacks and intrusion detection from network flow data of networking devices. For designing model an open source network flow based dataset CICIDS2017 is used. Results over XGBoost is compared with other algorithms which outperformed in case of CICIDS dataset and give detection rate of 91.61% and false alarm rate and attack misses were reduced up to 0.005% and 8.38% respectively.

ACKNOWLEDGMENT

We acknowledged Canadian Institutes of Cybersecurity and Sharafaldin et al. for providing us opportunity to use this useful and open source dataset for attacks detection purpose.

REFERENCES

[1] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), 20.

[2] Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. *International Journal of Engineering & Technology*, *7*(3.24), 479-482.

[3] Boukhamla, A., & Coronel, J. (2018). Cicids 2017 dataset: performance improvements and validation as a robust intrusion detection system testbed. *International Journal of Information and Computer Security*, *9*.

[4] Ustebay, S., Turgut, Z., & Aydin, M. A. (2018, December). Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier. In *2018 international congress on big data, deep learning and fighting cyber terrorism (IBIGDELFT)* (pp. 71-76). IEEE.

[5] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018, January). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP* (pp. 108-116).

[6] Ahmim, A., Maglaras, L., Ferrag, M. A., Derdour, M., & Janicke, H. (2019, May). A novel hierarchical intrusion detection system based on decision tree and rules-based models. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (pp. 228-233). IEEE.